Code Smell 61





**ABAP** 

Apex Apex

C C

C++

CloudFormation

COBOL COBOL

C# C#

**E** CSS

**⋈** Flex

**-co** Go

5 HTML

🐇 Java

JS JavaScript

Kotlin

Kubernetes

Objective C

PHP

PL/I PL/I

PL/SQL

Python

RPG RPG

Ruby

Scala

Swift

Terraform

**Text** 

TS TypeScript

T-SQL

**VB** VB.NET

VB6 VB6

XML XML



## Flex static code analysis

Unique rules to find Bugs, Security Hotspots, and Code Smells in your FLEX code

9

Security Hotspot 1

₩ Bug

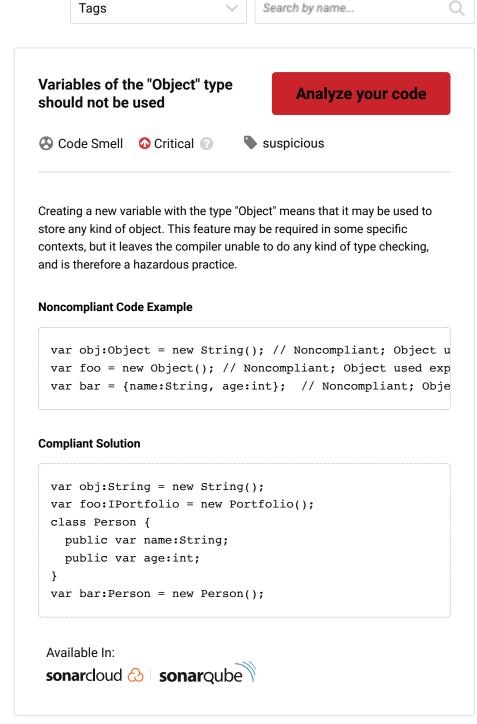
All rules 76 Sulnerability 5
Security.allowDomain() should only be used in a tightly focused manner
<b>⑥</b> Vulnerability
The flash.system.Security.exactSettings property should never be set to false
<b>⑥</b> Vulnerability
Dynamic classes should not be used  Code Smell
Code Smell
"LocalConnection" should be configured to narrowly specify the domains with which local connections to other Flex application are allowed
❸ Vulnerability
"default" clauses should be first or last
Code Smell
Event types should be defined in metadata tags
Code Smell
Event names should not be hardcoded in event listeners
Code Smell
The special "star" type should not be used
Code Smell
Variables of the "Object" type should not be used
☼ Code Smell
Methods should not be empty
Code Smell
Constant names should comply with a naming convention
Code Smell

All branches in a conditional structure should not have exactly the same

Classes that extend "Event" should

implementation

**Bug** 



© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved. Privacy Policy

override "Event.clone()"  • Bug
Constructors should not dispatch events
Rug
"ManagedEvents" tags should have companion "Event" tags
₩ Bug
Objects should not be instantiated inside a loop
Two branches in a conditional structure should not have exactly the same implementation
Constructor bodies should be as lightweight as possible
Only "while", "do" and "for" statements should be labelled
Statements, operators and keywords specific to ActionScript 2 should not be used
"for" loop stop conditions should be invariant
Unused function parameters should be removed
Code Smell